Research Project Proposal: Structured Learning

Alberto Archetti, alberto1.archetti@mail.polimi.it

1. Introduction to the problem

In recent years, deep learning models outperformed state-of-the-art techniques in many machine learning tasks, bringing ground-shaking advances in research and industrial environments. Deep learning models involve powerful function approximators to be trained over huge sets of data. As an example, the review [21] collects deep learning advances in common computer vision tasks, such as motion tracking, action recognition, human pose estimation, and semantic segmentation.

Neural networks are a computational tool that computer scientists know since the middle of the last century. They implement a "threshold logic", loosely inspired by biological neurons of the human brain. Despite their early invention, deep learning based on neural networks became popular only since 2012, when AlexNet [12], a deep convolutional neural network, won several computer vision competitions. This advance was due to the great improvements in GPU design, which sped significantly the training time of the networks. Deep learning techniques are heavily used today in research and industry because they build data-driven features from data. Before deep learning, features were hand-crafted by domain experts. This approach was applicable up to a satisfactory rate of success only in some areas and not feasible in others. The ability of building meaningful features in an automated way is extremely powerful, as we do not need to rely on domain knowledge and we can discover meaningful patterns without any human supervision.

Even if the literature is overflowing with incredibly successful applications of deep learning architectures based on neural networks, they come with some structural drawbacks. The first one is explainability [7]: it is very difficult to delve into their complex structure and reconstruct the *reasoning*, in a human-understandable sense, that brings the network to a specific decision. Explainability is mandatory in large-scale, safety-critical and mission-critical applications, as requirements impose hard constraints on model verifiability. The second drawback is the size of training set necessary for the learning procedure, called backpropagation. Since deep neural networks have a huge set of parameters, their tuning requires a high cost in terms of data, computational time, and memory.

Recently, many researchers started experimenting with embedding algorithmic priors directly in the neural network structure. Their aim is to insert algorithmic modules with a predefined behavior in the architecture while maintaining the end-to-end training procedure. This approach increases explainability, as algorithmic priors force the network to adapt to the algorithms' interfaces. It also helps training, because it reduces the variance of the model by lowering the number of tunable parameters. Finally, it provides a stabler solution during training, being it theoretically guided. This emerging field is called Structured Learning: algorithms provide structure and neural networks provide flexibility to learn and adapt from data.

The backpropagation algorithm, in order to train a structured system end-to-end, requires all the architecture components to be differentiable. End-to-end training is a desirable property, as it requires a single input-output dataset. A non-end-to-end training process would require separate tuning of each architectural component. This approach may be costly or not feasible in general. So, in order to embed an algorithm in a deep neural architecture and maintain end-to-end learning, it must be differentiable. This property can be achieved by using automatic differentiation, a technique common in fluid dynamics or engineering design optimization, but still underused in the machine learning area. Many experiments show the potential of Structured Learning in deep learning architectures and compiler experts are moving towards providing automatic differentiation as a default feature of programming languages.

2. Main related works

Algorithmic prior embedding is becoming an important tool in deep learning research. It is commonly used in computer vision with differentiable renderers for 3D reconstruction [4, 11, 13, 14, 16, 19, 23]. Some other fields in which it started to spread include physics simulations with differentiable physics engines [5, 18], automation and control [1, 10], and logical reasoning [22, 24].

The algorithmic embedding is often based on automatic differentiation (AD) [3, 15]. This technique is at the center of the differentiable programming research (∂P) [9, 20], which is aimed at providing differentiation as a first-class feature of programming languages. However, AD is not the only attempt of differentiating soundly general algorithms. In [17], the authors propose a differentiation model that takes into account the control flow of the program and not only the execution trace, as AD-based techniques do.

We can classify Structured Learning according to two dimensions: the first is the area of the task for which it is applied and the second is the architecture type. There are two main architectures, Structured Autoencoders and Structured Pipelines. The first is based on classical autoencoders, with an algorithmic decoder instead of a neural one. It is well suited for inverse problems, such as parameter estimation in physics or rendering in computer vision. The second is more general and consists of the composition of several modules, both neural and algorithmic. It is usually trained in a supervised way and has proven to be able to handle complex tasks, like simulated robotic motion and solution of visual Sudoku, all by end-to-end training.

The experiments that involve Structured architectures show improvements in data efficiency and performance with respect to unstructured deep neural networks. The main issue identified is the increase in computational complexity during training.

3. Research plan

3.1. Goal

The advantages provided by Structured Learning in explainability, data efficiency and solution structure with respect to deep unstructured networks are pushing research towards algorithmic prior integration. However, the study of algorithmic differentiability with non-AD techniques is in a very early stage and we did not find in the literature any attempt at integrating a Structured Learning architecture in a complex, real, environment. Our contribution is to explore the strengths and boundaries of Structured Learning from a technological aspect and implement a Structured Pipeline for a robotic agent as case-study. The challenge is to show the feasibility of integrating the model-based approach in a model-free setting. In detail, the research activity will start from the study of the differentiation properties that best suit our application, then we will choose an architecture for the system. We will implement and test it in a simulated environment and possibly integrate it in a real situation. We want to provide results that compare our architecture with respect to the current standards, focusing on the performance, explainability and data efficiency aspects. Lastly, we will discuss the current technological limits of Structured Learning integration.

The task we want to takle is autonomous navigation of a robotic agent. This task is particularly interesting for several reasons. First, research has produced a huge corpus of algorithmic techniques for solving tasks such as vision, perception, state estimation, obstacle detection, planning, control, multi-agent coordination and many more. The goal of Structured Learning is to integrate these techniques into deep end-to-end architectures, in order to bridge model-based and model-free design. Autonomous navigation is also relevant to many industrial applications. Autonomous cars, as a flourishing case-study today, aim at revolutionizing the way in which we perceive transportation, by granting efficient and secure performance. Since 2004 DARPA Grand challenge, major automotive industries invested billions of dollars in research and technological development around autonomous driving. However, autonomous driving is a safety-critical application and the integration of deep learning techniques is still open to debate. The potential contribution of Structured Learning is to alleviate both the explainability issues in fault scenarios and the data hunger of current architectures.

3.2. Research activities

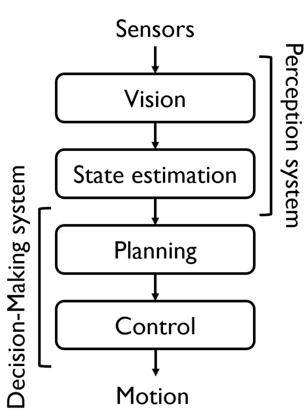
1. Differentiation inquiry Algorithm differentiation is at the core of Structured Learning. The first step of our research is to analyze and compare AD-based techniques with non-AD-based techniques for algorithmic differentiation. AD-based techniques have a strong technological support, but may not provide a correct derivative at boundary conditions for control flow. Non-AD-based techniques aim at solving this issue, but they may lack generality, a well-established corpus of studies or technological support. The analysis will mainly focus on algorithms for robotic motion that we can potentially implement in the architecture.

2. Architecture design It is difficult to select a standard reference architecture for autonomous navigation. Autonomous driving surveys [2, 8], for example, suggest two macro modules, *Perception System* and *Decision-Making System*, that are further divided into many submodules. Since the main focus of this work is to further study the applicability of Structured Learning and not to develop a full software stack for autonomous driving, we will focus on a simpler architecture of four modules, *Vision*, *State estimation*, *Planning* and *Control* (Figure a). This architecture captures the main aspects of navigation: detect and analyze the surroundings, provide an estimation of the current state of the robot with respect to the environment, plan towards a predefined goal and provide to the actuators the necessary inputs for a correct motion.

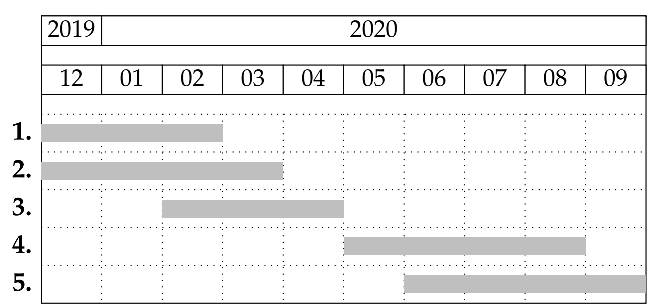
3. Implementation and training Given the differentiation and architectural requirements, we must choose the best suited machine learning framework and implement the system. Then, we must gather the training data from the available datasets or collect it ourselves and train the architecture. The feedback with the architecture design activity is very important. Architectural choices may affect the framework selection, but the features provided by the available framework may limit the options in the previous stage.

4. Testing We test the trained system in a simulated environment e.g. CARLA Simulator [6] and compare the performance with other frameworks from the state-of-the-art of autonomous navigation. Our ultimate goal, however, is to integrate the system in a real environment. As far as we know, experiments in a real-life scenario involving Structured Learning techniques have never been conducted. It is interesting to test the resilience of the system to noise and non-idealities of the real world.

5. Conclusions We provide a report of the experiments we've conducted, highlighting the strengths and weaknesses of our architecture. At this point, we can indicate the direction of future research that need to be investigated further. This step is interleaved with the testing activity, and coincides to the final paper redaction. We expect to conclude this work by September 2020 (Figure b).



(a) Simplified architecture of an autonomous navigation system, inspired by [10]



(b) Gantt chart of the research stages

References

[1] Amos, B., Jimenez, I., Sacks, J., Boots, B., and Kolter, J. Z. Differentiable mpc for end-to-end planning and control. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8289–8300.

[2] Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T., Mutz, F., Veronese, L., Oliveira-Santos, T., and Souza, A. F. D. Self-driving cars: A survey, 2019.

[3] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey, 2015.

[4] Che, C., Luan, F., Zhao, S., Bala, K., and Gkioulekas, I. Inverse transport networks, 2018.

[5] de Avila Belbute-Peres, F., Smith, K., Allen, K., Tenenbaum, J., and Kolter, J. Z. End-to-end differentiable physics for learning and control. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 7178–7189.

[6] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning* (2017), pp. 1–16.

[7] Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. Explaining explanations: An overview of interpretability of machine learning, 2018.

[8] Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics* (Nov 2019).

[9] Innes, M., Edelman, A., Fischer, K., Rackauckas, C., Saba, E., Shah, V. B., and Tebbutt, W. A differentiable programming system to bridge machine learning and scientific computing. *ArXiv abs/1907.07587* (2019).

[10] Karkus, P., Ma, X., Hsu, D., Kaelbling, L. P., Lee, W. S., and Lozano-Perez, T. Differentiable algorithm networks for composable robot learning, 2019.

[11] Kato, H., Ushiku, Y., and Harada, T. Neural 3d mesh renderer. *CoRR abs/1711.07566* (2017).

[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[13] Liu, S., Li, T., Chen, W., and Li, H. Soft rasterizer: A differentiable renderer for image-based 3d reasoning, 2019.

[14] Loper, M. M., and Black, M. J. Opendr: An approximate differentiable renderer. In *Computer Vision – ECCV 2014* (Cham, 2014), D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Springer International Publishing, pp. 154–169.

[15] Margossian, C. C. A review of automatic differentiation and its efficient implementation. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (Mar 2019), e1305.

[16] Petersen, F., Bermano, A. H., Deussen, O., and Cohen-Or, D. Pix2vex: Image-to-geometry reconstruction using a smooth differentiable renderer. *CoRR abs/1903.11149* (2019).

[17] Petersen, F., Borgelt, C., and Deussen, O. Algonet: C^{∞} smooth algorithmic neural networks. *CoRR abs/1905.06886* (2019).

[18] Seo, S., and Liu, Y. Differentiable physics-informed graph networks, 2019.

[19] Tewari, A., Zollhöfer, M., Kim, H., Garrido, P., Bernard, F., Pérez, P., and Theobalt, C. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction, 2017.

[20] van Merrienboer, B., Breuleux, O., Bergeron, A., and Lamblin, P. Automatic differentiation in ml: Where we are and where we should be going. In *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 8757–8767.

[21] Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience 2018* (02 2018), 1–13.

[22] Wang, P., Donti, P. L., Wilder, B., and Kolter, J. Z. Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. *CoRR abs/1905.12149* (2019).

[23] Wu, S., Rupprecht, C., and Vedaldi, A. Photo-geometric autoencoding to learn 3d objects from unlabelled images, 2019.

[24] Yang, F., Yang, Z., and Cohen, W. W. Differentiable learning of logical rules for knowledge base reasoning. In *NIPS* (2017).